

La prova si svolge a libri chiusi (non è permessa la consultazione di materiale didattico).

Traccia delle soluzioni

Si consideri il caso di studio PolliceVerde e in particolare:

**Caso d'uso: CreaRichiestaIntervento.**

Breve descrizione: *Permette, una volta autenticatisi, di compilare la SL. Nel caso di scheda creata dalla filiale, permette di accettare la SL, altrimenti di inviare mail alla filiale.*

Attore principale: *Compilatore*

Attore secondario: *MailServer*

PreCondizioni: *Compilatore Autenticato*

PostCondizioni: *Scheda lavoro creata. Se creata da filiale, accettata, altrimenti mail inviata alla filiale.*

Sequenza principale degli eventi:

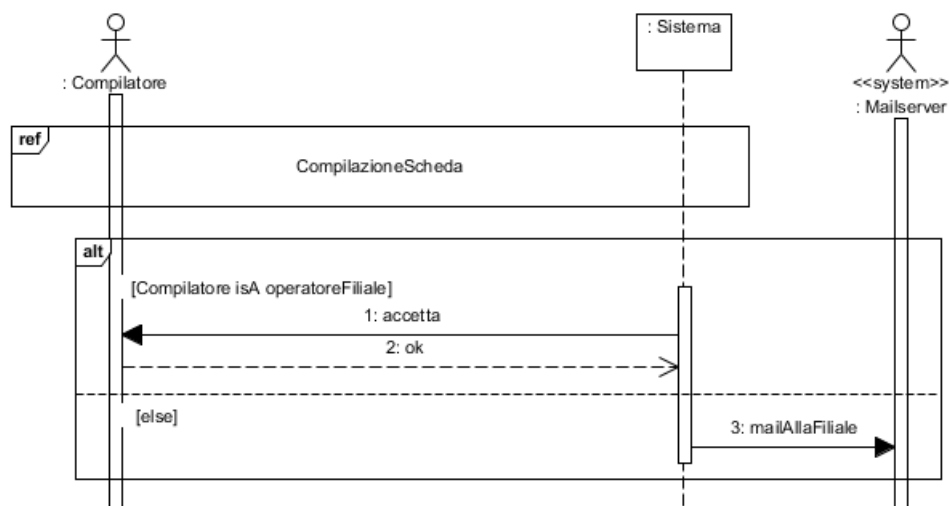
- 1 **include** *CompilazioneScheda*
- 2 **if** (*Compilatore = OperatoreFiliale*)
  - 2.1 *Il sistema richiede l'accettazione*
  - 2.2 *L'operatoreFiliale accetta la richiesta*
- 3 **altrimenti**
  - 3.1 *Il sistema genera una email e la invia alla filiale*

Sequenza alternativa degli eventi: *Ulteriori modifiche.*

**Domanda 1.** Realizzazione dei casi d'uso.

Assumendo di aver definito il diagramma di sequenza *CompilazioneScheda* che descrive il caso d'uso *CompilazioneScheda*, dare un diagramma di sequenza che realizza il caso d'uso **CreaRichiestaIntervento** (solo sequenza principale).

**Risposta.** Un possibile diagramma di sequenza è il seguente.



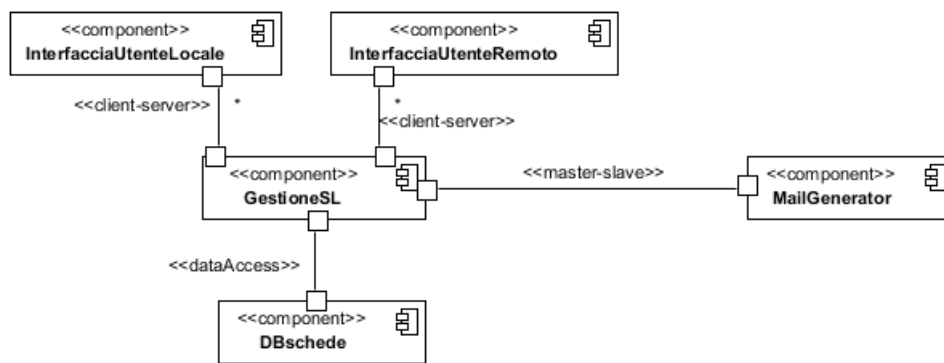
\*\*\*\*\*

Per realizzare la gestione delle schede lavoro sono state individuate le seguenti componenti:

Componente	Responsabilità
InterfacciaUtenteLocale	Permette a un operatore della filiale di interagire col sistema, in particolare per creare e compilare una richiesta intervento.
InterfacciaUtenteRemoto	Permette a un cliente e/o un giardiniere di interagire col sistema, in particolare per creare e compilare una richiesta intervento.
GestioneSL	Responsabile della gestione delle schede lavoro.
DBschede	Mantiene le schede lavoro.
MailGenerator	Genera e invia al MailServer le mail di richiesta di valutazione.

**Domanda 2.** Architettura. Fornire una vista C&C delle componenti introdotte.

**Risposta.**



\*\*\*\*\*

L'analisi dei requisiti e la progettazione hanno portato a ulteriori scelte: chi compila una SL in remoto si collega al sistema usando un'interfaccia web; i dipendenti della filiale lavorano su un Pc.

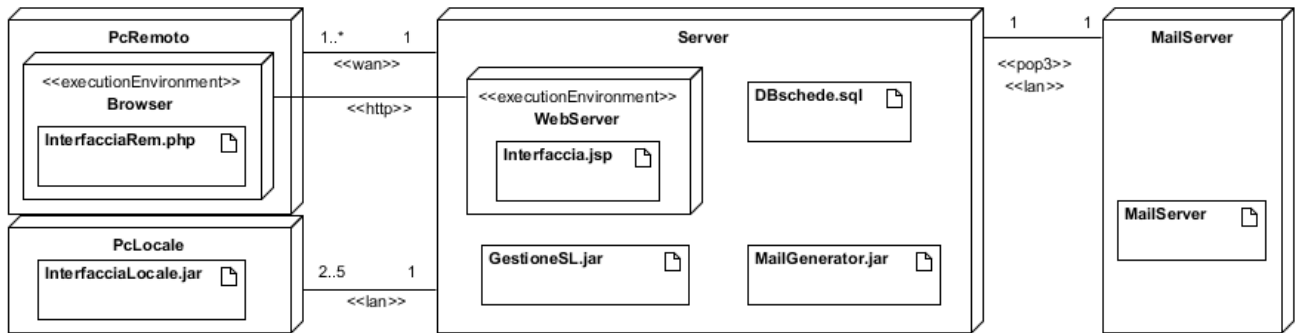
Gli artefatti che manifestano le componenti introdotte sopra sono:

- *InterfacciaLocale.jar* manifesta l'interfaccia per l'utente locale sul suo Pc;
- *InterfacciaRem.php*, visualizzato da un browser di una macchina client e *Interfaccia.jsp*, dislocata su un web server, manifestano l'interfaccia remota;
- *GestioneSL.jar*, *DBschede.sql*, *MailGenerator.jar*, dislocate su una macchina server.

Si assuma inoltre che il *MailServer* sia su una macchina dedicata, esterna al sistema da realizzare, di tipo *ServerMail*, collegata in rete locale con il server.

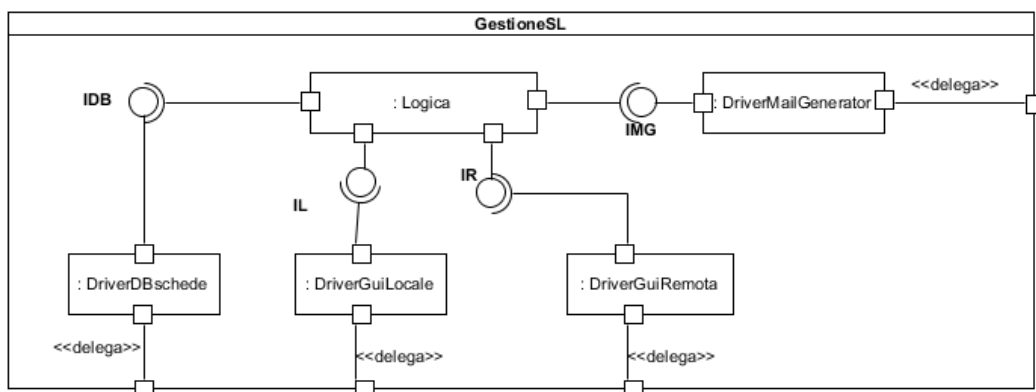
**Domanda 3.** Architettura: Fornire una vista di dislocazione dell'architettura della parte di sistema di interesse, più il server di posta, esplicitando anche le molteplicità delle connessioni.

**Risposta**



**Domanda 4.** Progettazione di dettaglio. Dare un diagramma di struttura composta per la componente GestioneSL. Di che tipo sono i connettori che collegano i driver ai porti della componente? Indicarlo nel diagramma introducendo uno stereotipo.

**Risposta.**



\*\*\*\*\*

Il metodo, da usare a fine mese per archiviare le schede lavoro di interventi chiusi, differenzia le informazioni salvate a seconda se si tratta di

- a) intervento inAbbonamento,
- b) interventi extraAbbonamentoSottoSoglia,
- c) interventi extraAbbonamentoOltreSoglia.

Le informazioni vengono salvate rispettivamente in forma “ridotta”, “normale”, “estesa”.

Il seguente metodo determina la forma in cui devono essere salvate le informazioni:

```
public String[] formaArchiviazione (Intervento[] listaInterventi);
```

Senza entrare nel dettaglio delle SL, indicheremo con interX gli interventi, con la convenzione che X sia un intero, e che valori di X nella prima decade (tra 0 e 9) indichino interventi di tipo a), quelli nella seconda decade interventi di tipo b), e quelli nella terza di tipo c).

**Domanda 5.** (Verifica)

Si completi con i valori attesi il seguente insieme di casi di test, in cui utilizziamo la notazione Java per l'inizializzazione degli array, per indicare un valore di tipo array:

a)

input	Output
{}	
{inter9, inter10, inter19}	

b) Si consideri la seguente realizzazione di formaArchiviazione:

```
public String [] formaArchiviazione(Intervento[] listaInterventi) {
    String [] lia = new String[listaInterventi.length]; //1
    for(int i=0; i < listaInterventi.length; i++) {
        Intervento intervento = listaInterventi[i]; //2
        TipoIntervento tipo = intervento.tipo(); //2
        if (tipo == "inAbbonamento"){ //3
            lia[i] = "normale"; //3
        } else if (tipo == "extraAbbonamentoOltreSoglia") { //4
            lia[i] = "ridotta"; //4
        } else { //5
            lia[i] = "estesa"; //5
        }
    }
    return lia; //6
}
```

Dare un diagramma di attività per il grafo di flusso del metodo, utilizzando i numeri a sinistra come nomi dei nodi.

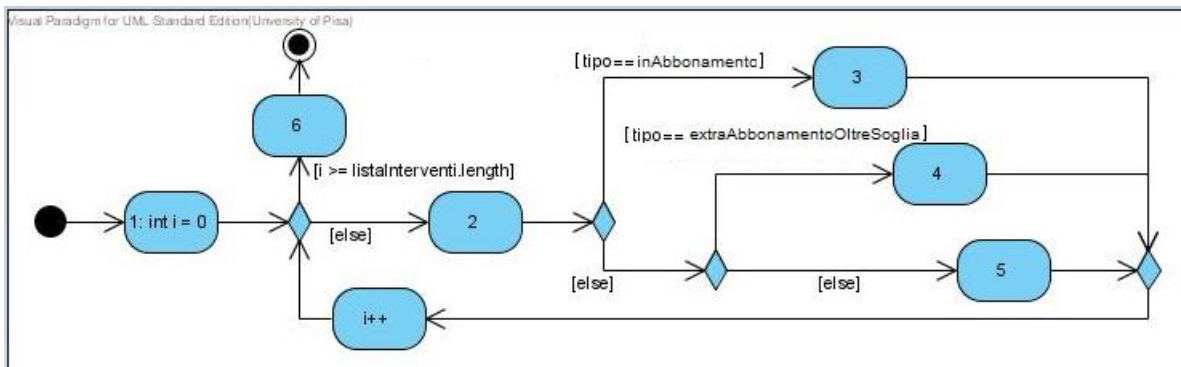
c) Cosa si può dire sulla copertura del codice ottenuta eseguendo i casi di test dati in a), rispetto al criterio delle decisioni? Giustificare la risposta.

**Risposta:**

a)

Input	output
{}	{}
{inter9, inter10, inter19}	{"ridotta", "normale", "normale"}

b)



c) Dei 6 archi uscenti da decisioni, non viene percorso quello che porta al nodo 4, quindi la copertura è di 5 su 6, ovvero dell'83 % circa.